BERTweetConvFusionNet: Enhancing BERTweet for Twitter Text Sentiment Classification

Debeshee Das, Mariia Eremeeva, Laura Schulz, Piyushi Goyal
{debdas,emariia,laschulz,pgoyal}@student.ethz.ch
Group : scrum master
Department of Computer Science, ETH Zurich, Switzerland

***Abstract*—The vast and continuous production of unstructured data by social media platforms like Twitter necessitates efficient automated analysis techniques for sentiment analysis. The brevity and colloquial language of Tweets pose multiple challenges to traditional opinion mining algorithms. The large volume of text makes the acquisition of high-quality, human-annotated training data nearly impossible. Hence, this paper proposes a novel deep-learning based solution for Twitter sentiment analysis that addresses the challenges of automatic and noisily labelled data. Leveraging the pre-trained *BERTweet* model for embeddings, we develop a novel CRNN-based 'fusion net' architecture combining CNN, RNN, and Attention layers. Through a systematic exploration of various neural network, pre-processing, and ensembling configurations, our approach achieves an accuracy of 92.12% on the public leaderboard.**

I. Introduction

Social media platforms like Twitter generate copious amounts of unstructured data every second. When analyzed effectively, these extensive data streams can yield significant insights into improved marketing strategies for brands, public opinion, social issues, and the mitigation of hate speech [1]. Specifically, sentiment analysis of Twitter data can assist market participants in comprehending their audience or customer base through social media interactions [2].

The large scale of this data however, requires efficient and automated analysis techniques to be feasible. Automated analysis of micro-blogging platforms such as Twitter is challenging because the text snippets are usually very short (strict 140-character limit) and use colloquial language, causing traditional opinion mining algorithms to perform poorly [3]. In addition, the huge volume makes it nearly impossible to get human-annotated or good quality training data. Hence, many previous works have resorted to working with noisy automatic labelling. Similarly, we work with automatically labelled (distant supervision) data by selecting tweets that contain ':)' for positive, and ':(' for negative sentiment data samples [4]. We can easily see how sources of noise such as sarcasm, irony, and mixed sentiments can creep into the dataset using this labelling technique.

In this paper, we set out to develop a performant and novel solution for the challenging task of Twitter sentiment classification, given a noisily labelled dataset. We first explore a few baselines using classical machine learning (ML) methods such as logistic regression, support vector machines and random forest models on various types of embeddings of the text snippets. Next, we deep-dive into deep learning methods. In particular, we explore the effectiveness of the pre-trained *BERTweet* model by fine-tuning it and evaluating its performance on a held out validation set.

Next, we investigate Convolutional Recurrent Neural Networks (CRNNs) as they have been successful at NLP tasks such as sequence labelling, text classification and sentiment classification [5], [6]. We develop our novel CRNN-based 'fusion net' architecture on the *BERTweet* embeddings of the text snippets. We systematically explore various combinations of CNN, RNN and Attention layers for this 'fusion net', to identify the best performing composition. Our novel architecture, *BERTweetConvFusionNet*, boosts accuracy and obtains a high classification accuracy of 92.12% on the public leaderboard test set when ensembled with fine-tuned *BERTweet* models (with and without processing), despite the noise in the dataset.

II. Related Work

Pang et al. [7] demonstrated that standard ML approaches like Naive Bayes, maximum entropy classification, and support vector machines underperformed in sentiment classification compared to topic-based classification, highlighting the complexity of capturing sentiment. A survey of traditional ML techniques for sentiment analysis [8] confirmed the need for a language model that captures complex contextual information. For this, we turn to advancements in natural language processing (NLP) using deep learning.

A most important precursor to modern state-of-the-art solutions to NLP tasks is the Bidirectional Encoder Representations from Transformer model (BERT) [9]. BERT was elegantly designed to be pre-trained on representations from unlabelled text by jointly conditioning on both left and right contexts and then fine-tuned with an additional output layer to create state-of-the-art models for a wide range of tasks. Leveraging the versatile nature of BERT, [10] obtained 92% accuracy on sentiment analysis on a real-world dataset, while [11] fine-tuned BERT by adding a single adapter (output) layer. [12] took it a step further by modifying BERT's architecture to employ a transformer

model with more emphasis on encoders that scan the total series of words all at once, obtaining an accuracy of 96% on the Kaggle SMILE dataset [13].

Pre-trained models have become a cornerstone in the field of NLP [14]. They provide a better starting point for training on downstream tasks and often lead to improved generalization and faster convergence compared to training models from scratch. *BERTweet* [15] is a public large scale pre-trained language model for English tweets. A promising approach to sentiment analysis is extracting the embeddings from a deep learning model like BERT and then training another deep neural network. Various works have experimented with convolutional neural networks (CNNs) and recurrent neural networks (RNNs), but the best results seem to stem from a fusion of both [16]. [6] applies this concept to the sentiment analysis problem by training a neural net composed of convolutional layers and a BiLSTM on BERT model embeddings of the inputs. Concluding this section, we refer the readers to the most up to date extensive survey on sentiment analysis from social media platforms [17] where the authors explore new aspects of sentiment analysis such as temporal dynamics and applications in industry.

III. Data and its Pre-processing

Since we are working with an automatically labelled dataset for sentiment analysis with sources of noise such as sarcasm and ambiguity, it is important to investigate any possible pre-processing measures that could help mitigate such noise. First, we analyzed the provided training dataset: examining the lengths of positive and negative tweets, comparing the number of hashtags in each, and extracting the number of tweets containing symbol-based emoticons and abbreviations. We found more positive hashtags (over 200k) compared to negative hashtags (under 150k), with 11.8% of tweets containing hashtags. Only 0.1% of tweets had symbol-based emoticons, while 17.9% used abbreviations. Considering these results, we developed a *basic pre-processing regime* that includes removing tweets present in both positive and negative training data to address noise due to ambiguity, removing repeated punctuation and symbols, and replacing $\langle user \rangle$ with empty strings, and $\langle url \rangle$ with "HTTPURL" to comply better with pre-trained BERT models. Multiple additional pre-processing steps were applied to both the baseline *BERTweet* model and our novel *BERTweet-ConvFusionNet* and compared (see Appendix Table III). Although pre-processing measures seemed to downgrade the performance of individual models, they boosted the performance of ensembled models (see Section VI).

IV. Baselines

A. Classical ML Methods

We initially experimented with traditional ML techniques, including Random Forest, Linear SVC, and Logistic Regression. Following the basic pre-processing described in Section III, we tested various embedding types such as Sentence Embeddings, TF-IDF, Word2Vec, and Bag of Words. The combinations that achieved the highest validation accuracy are detailed in Appendix Table IV. The best performance on a held-out 10% validation set was 77.58%, which did not exceed the leaderboard baseline. Hence, we proceeded to exploring deep learning approaches instead.

B. Deep-Learning Approaches

Pre-trained models provide an excellent starting point for our setting because they allow for the efficient usage of large training datasets enhanced generalization and prevention of overfitting [18], and often lead to state-of-the-art performance on many similar tasks [19]. Hence, we first explored various settings for fine-tuning the pre-trained *BERTweet* model [15] (`BERTweet-base` model with 135 million parameters), to serve as a baseline model. For all fine-tuning experiments, we retained the model architecture (no additional parameters added). We trained on the entire training dataset of 2.5 million data points using a batch size of 64 and early stopping to prevent over-fitting. We experimented with various settings of pre-processing, particularly making sure to align with the data pre-processing used for pre-training *BERTweet*, and identified the best pre-processing for the model, as reported in Appendix Table III.

V. Methodology

In this section, we describe how we systematically constructed our *BERTWeetConvFusionNet* architecture, starting with the pre-trained *BERTweet* model.

A. Enhancing BERTweet

The hybrid architecture of CRNNs leverages the strengths of both CNNs and RNNs to handle tasks involving spatial and temporal dependencies. Before developing our novel CRNN-based 'fusion net', we first experimented with various BERT model embeddings, such as DistilBERT and DistilRoBERTa [20]. The pre-trained *BERTweet* model unsurprisingly performed the best (see Appendix Table V). Hence, we used *BERTweet* embeddings for all our further experiments. For a default base model, we fixed three CNN-layers, followed by an LSTM layer with a dropout rate of 0.2 and then tested the following variations:

1) *Replacing the 2D-CNN with 1D-CNN*: Inspired from [21], where it has been shown that specifically in the case of sentence classification, simplifying the model and reducing computational complexity can achieve excellent results on multiple benchmarks, we replaced the 2D-CNN layers with 1D-CNN layers.
2) *Varying the number of CNN layers*: Introducing very deep convolutional networks [22] is associated with improvements over the state-of-the-art on several public text classification tasks. Hence, we experiment with up to five CNN layers.

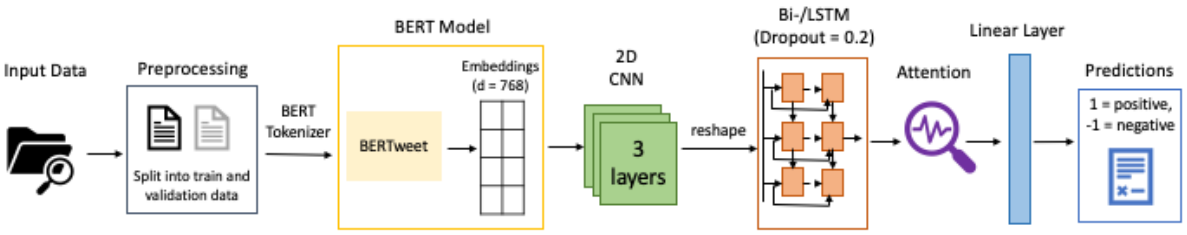

Figure 1: The architecture of *BERTweetConvFusionNet*

3) *Comparing Bi-LSTM and LSTM*: While bi-directional LSTMs [23] can capture information from both past and future states, uni-directional LSTMs simplify the architecture by only focusing on past states. We explored both variations to understand their impact on capturing temporal dependencies in text data and if the added complexity translates to better performance.
4) *Introducing an Attention layer*: Attention [24] helps the model to focus on the most relevant parts of an input sequence and often leads to improvements of interpretability and performance. Building on previous work in text sentiment classification [25], [26], we add an Attention layer after the RNN layer as it enables the model to attend to specific parts of the sequence that are most informative for the task [27].

B. Our Novel Architecture

Our novel *BERTweetConvFusionNet* model, shown in Figure 1, processes data through the pre-processing module described in Section III, splitting it into training and validation sets. The data is tokenized using the BERT tokenizer, and embeddings are extracted using the pre-trained *BERTweet* model. These embeddings pass through multiple 2D-CNN layers to capture n-gram features and local dependencies, then feed into an LSTM/BiLSTM layer with a dropout probability of 0.2 to capture temporal dependencies. An attention mechanism highlights the most relevant parts of the sequence, followed by a linear layer that transforms the attended features into final prediction probabilities, classifying the text as positive or negative sentiment. These probabilities are either directly converted into predictions or ensembled with other models before converting to prediction labels.

C. Ensembling Predictions

To benefit from the features learned by the various model architectures, we implement an ensemble at inference time. The predicted probabilities from each model are combined using a (equally by default) weighted average.

D. Training and Evaluation

For our initial experiments, we train using the given subset of $200,000$ training data points and for all final results reported in Table II, we use the entire training set of 2.5 million data samples. For all experiments, we carve out a fixed validation set (10% with the small training set and 5% on the full dataset) which we use to monitor the training. With validation accuracy as the metric, we implement early stopping with patience to avoid overfitting as shown in Figure 2. For all experiments we use a batch size of 64, learning rate of 2×10^{-5}, and weight decay 0.01.

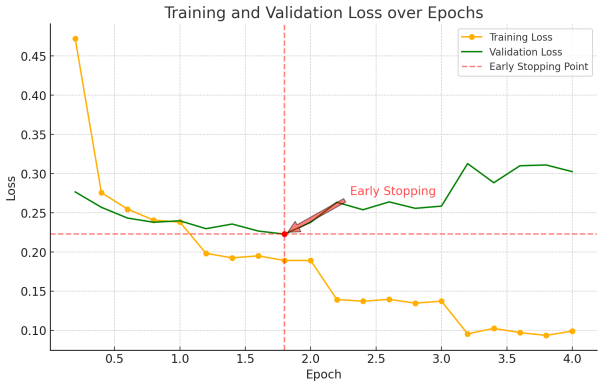

Figure 2: Monitoring training and validation loss while training with early stopping

VI. Experimental Results And Discussion

While developing our *BERTweetConvFusionNet* architecture, we experimented with various configurations (see Section V-A). Using a subset of $200,000$ datapoints for training, we reported validation accuracies in Table I. Final accuracies for our novel architectures, baselines, and ensembled models, trained on the full 2.5 million datapoints, are in Table II.

A. Architecture Experiments

Leveraging the pre-trained *BERTweet* model embeddings in various combinations, we found that removing the 2D-CNN-layers or reducing their complexity to 1-dimensional shows the biggest negative impacts (Table I Row 2). Hence, we use three 2D-CNN layers in our final architecture. We also find that a bi-directional LSTM shows a slightly better performance as compared to an LSTM (Table I Row 3), consistent with previous work [28]. Adding an attention

	Model	Validation Accuracy (%)
1	Embd + 2D-CNN	91.14
	Embd + LSTM	91.19
	Embd + Bi-LSTM	91.24
2	Embd + 1D-CNN + LSTM	91.21
	Embd + 2D-CNN (5layers) + LSTM	91.17
3	Embd + 2D-CNN + LSTM	91.27
	Embd + 2D-CNN + Bi-LSTM	91.29
4	Embd + 2D-CNN + LSTM + Attn.	91.28
	Embd + 2D-CNN + Bi-LSTM + Attn.	91.28

Table I. Classification Accuracies of various architectures evaluated using a 10% held out validation set. 'Embd' refers to *BERTweet* embeddings and a default configuration uses three CNN layers. 'Attn.' refers to the added attention layer.

layer did not conclusively improve the validation accuracy (Table I Row 4).

B. Ablation Study

We drew conclusions about the contribution of individual components by systematically removing them (Table I Row 1). We find that without any RNN layer, a plain 2D-CNN has the poorest performance. We also show that RNN layers, both LSTM and BiLSTM, without CNN layers have sub-par performance. Hence, the CRNN architecture of *BERTweetConvFusionNet* is critical to capture effective signals for sentiment classification.

C. Results

From Table II, we observe an average improvement of 0.5% in validation accuracy when training on the full dataset compared to the subset. Classical ML-based methods achieve a maximum accuracy of 77.6% on the public leaderboard, failing to clear the baseline. In contrast, the fine-tuned *BERTweet* model attains 91.64%. Notably, fine-tuning *BERTweet* with our best pre-processing achieved the highest validation score of 91.89% but did not generalize as well to the public leaderboard test set as the model without pre-processing. Although our *BERTweetConvFusionNet* model with Attention (no pre-processing) surpassed the *BERTweet* baseline in validation, the version without the Attention layer generalized better on the public leaderboard test set. We also report leaderboard classification accuracies after ensembling the test set predictions for the *BERTweet* model with and without pre-processing. Adding the three different configurations of *BERTweetConvFusionNet* one at a time to this ensemble led to improvements in the cases with BiLSTM configurations. Adding multiple versions of *BERTweetConvFusionNet*, with and without attention, resulted in the best ensembling configuration, achieving 92% on the test set, as shown in Table II. We repeated the best performing configuration of models with `BERTweet-large` (355 million parameters) instead of `BERTweet-base` which resulted in a performance improvement of 0.12%, giving us a final best score of 92.12% on the public leaderboard test set.

	Model	Validation Accuracy(%)	Leaderboard Accuracy(%)
(1)	RF with sentence embeddings	80.04	74.80
(2)	LR with sentence embeddings	79.89	77.60
(3)	BERTweet (no pre-processing)	91.76	91.64
(4)	BERTweet (best pre-processing)	91.89	91.36
(5)	Embd + 2D-CNN + LSTM + Attn.	91.86	91.44
(6)	Embd + 2D-CNN + Bi-LSTM + Attn.	91.79	91.40
(7)	Embd + 2D-CNN + Bi-LSTM	91.78	91.64
	Ensembles		
	(3) + (4)	-	91.88
	(3) + (4) + (5)	-	91.76
	(3) + (4) + (6)	-	91.96
	(3) + (4) + (7)	-	91.98
	(3) + (4) + (6) + (7)	-	92.00
	(3) + (4) + (6) + (7) + Large	-	**92.12**

Table II. Classification accuracy scores of our *BERTweet-ConvFusionNet* model compared with baselines and best performing ensembles. 'Large' refers to the use of `BERTweet-large` instead of `BERTweet-base`.

D. Discussion

Our results show that pre-processing data for *BERTweet* alone does not significantly improve performance. However, ensembling predictions from both preprocessed and raw datasets boosted accuracy, underscoring the value of combining data treatments. Interestingly, attention layers did not enhance individual models but improved performance in ensembles. The best results came from combining multiple configurations of *BERTweetConvFusionNet* with *BERTweet*-based models, indicating our novel architecture addressed challenging test samples that the base models missed.

VII. Conclusion

In this study, we demonstrated the potential of enhancing the *BERTweet* model for Twitter sentiment classification on a noisy dataset. Through systematic experimentation, we developed our novel *BERTweetConvFusionNet* architecture, which combines 2D-CNN layers with a BiLSTM/LSTM followed by an optional Attention layer. Multiple configurations of *BERTweetConvFusionNet* when ensembled with the base model *BERTweet* (with and without pre-processing), outperform all baselines and attain an accuracy of 92.12% on the public leaderboard. This improvement is evident despite the inherent noise in the dataset derived from distant supervision. Our approach leverages the strengths of pre-trained BERT model embeddings, convolutional and recurrent neural networks to capture local and sequential patterns and context in the text data, leading to more accurate sentiment predictions.

REFERENCES

[1] V. Sahayak, V. Shete, and A. Pathan, "Sentiment analysis on twitter data," *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, vol. 2, no. 1, pp. 178–183, 2015.

[2] A. Sarlan, C. Nadam, and S. Basri, "Twitter sentiment analysis," in *Proceedings of the 6th International Conference on Information Technology and Multimedia*, 2014, pp. 212–216.

[3] P.-W. Liang and B.-R. Dai, "Opinion mining on social media data," in *2013 IEEE 14th International Conference on Mobile Data Management*, vol. 2, 2013, pp. 91–96.

[4] J. Read, "Using emoticons to reduce dependency in machine learning techniques for sentiment classification," in *Proceedings of the ACL Student Research Workshop*, C. Callison-Burch and S. Wan, Eds. Ann Arbor, Michigan: Association for Computational Linguistics, Jun. 2005, pp. 43–48. [Online]. Available: https://aclanthology.org/P05-2008

[5] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," *arXiv preprint arXiv:1603.01354*, 2016.

[6] S. Kokab, S. Asghar, and S. Naz, "Transformer-based deep learning models for the sentiment analysis of social media data," *Array*, vol. 14, p. 100157, 04 2022.

[7] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*. Association for Computational Linguistics, Jul. 2002, pp. 79–86. [Online]. Available: https://aclanthology.org/W02-1011

[8] B. Pang, L. Lee *et al.*, "Opinion mining and sentiment analysis," *Foundations and Trends® in information retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[10] A. Chiorrini, C. Diamantini, A. Mircoli, D. Potena *et al.*, "Emotion and sentiment analysis of tweets using bert." in *Edbt/icdt workshops*, vol. 3, 2021, pp. 1–7.

[11] A. Samir, S. M. Elkaffas, and M. M. Madbouly, "Twitter sentiment analysis using bert," in *2021 31st international conference on computer theory and applications (ICCTA)*. IEEE, 2021, pp. 182–186.

[12] S. Mann, J. Arora, M. Bhatia, R. Sharma, and R. Taragi, "Twitter sentiment analysis using enhanced bert," in *Intelligent Systems and Applications: Select Proceedings of ICISA 2022*. Springer, 2023, pp. 263–271.

[13] "Smile twitter emotion dataset," Available at https://www.kaggle.com/datasets/ashkhagan/smile-twitter-emotion-dataset.

[14] K. Sun, X. Luo, and M. Y. Luo, "A survey of pretrained language models," in *International Conference on Knowledge Science, Engineering and Management*. Springer, 2022, pp. 442–456.

[15] D. Q. Nguyen, T. Vu, and A. T. Nguyen, "Bertweet: A pre-trained language model for english tweets," *arXiv preprint arXiv:2005.10200*, 2020.

[16] Z. Min, "Drugs reviews sentiment analysis using weakly supervised model," in *2019 IEEE international conference on artificial intelligence and computer applications (ICAICA)*. IEEE, 2019, pp. 332–336.

[17] M. Rodríguez-Ibánez, A. Casánez-Ventura, F. Castejón-Mateos, and P.-M. Cuenca-Jiménez, "A review on sentiment analysis from social media platforms," *Expert Systems with Applications*, vol. 223, p. 119862, 2023.

[18] X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huo, J. Qiu, Y. Yao, A. Zhang, L. Zhang *et al.*, "Pre-trained models: Past, present and future," *AI Open*, vol. 2, pp. 225–250, 2021.

[19] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Science China technological sciences*, vol. 63, no. 10, pp. 1872–1897, 2020.

[20] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[21] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint arXiv:1510.03820*, 2015.

[22] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," *arXiv preprint arXiv:1606.01781*, 2016.

[23] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks : the official journal of the International Neural Network Society*, vol. 18, pp. 602–10, 07 2005.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[25] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Knight, A. Nenkova, and O. Rambow, Eds. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 1480–1489. [Online]. Available: https://aclanthology.org/N16-1174

[26] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, J. Su, K. Duh, and X. Carreras, Eds. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 606–615. [Online]. Available: https://aclanthology.org/D16-1058

[27] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, 2016, pp. 207–212.

[28] S. Kashid, K. Kumar, P. Saini, A. Dhiman, and A. Negi, "Bi-rnn and bi-lstm based text classification for amazon reviews," in *International conference on deep learning, artificial intelligence and robotics*. Springer, 2022, pp. 62–72.

APPENDIX
PRE-PROCESSING EXPERIMENTS

Pre-processing	BERTweet	BERTweetConvFusionNet
No pre-processing	90.94	91.28
Basic	90.92	90.92
Basic with USER	90.68	90.86
Basic with abbrev	90.90	90.88
Basic with stemming	90.48	90.75
Basic with emoticon	90.93	91.00
Basic with hashtag segmentation	90.86	91.09

Table III. Accuracy scores on a 10% held-out validation set using various pre-processing techniques on *BERTweet* and our novel *BERTweet*ConvFusionNet model

In addition to the basic pre-processing steps described in Section III, we try the following techniques based on heuristic reasoning:

- *Hashtag Segmentation:* This breaks up the hashtag and appends those words with the tweet so that it helps in capturing the sentiment in the individual words.
- *Replace* $\langle user \rangle$ *with @USER:* This is done specifically to align our pre-processing with that of the data used for pre-training *BERTweet*.
- *Replace abbreviations with the expansion:* Since 17.9% of tweets use abbreviations, we expand them to try and capture the context in the individual words.
- *Replace symbol-based emoticons with the corresponding emotion-word:* Even though tweets with such emoticons were very rare, we find that they capture important sentiment related information.
- *Stemming:* We stem the words in a tweet so that the actual meaning can be captured better.

We report the empirical impacts of these additional pre-processing steps on classification accuracy (see Table III) by training both the *BERTweet* baseline model and our *BERTweetConvFusionNet* model with a 2DCNN + LSTM + Attn. configuration. Our findings indicate that additional pre-processing generally degrades the performance of both models. Among the various pre-processing techniques evaluated, only the replacement of symbol-based emoticons with corresponding words improved the performance of *BERTweet*. Consequently, this approach, termed 'Basic with emoticon', is identified as the best pre-processing configuration for the *BERTweet* model. Since pre-processing did not enhance the *BERTweetConvFusionNet* model, subsequent experiments with this model were conducted without any pre-processing.

CLASSICAL ML BASELINES

We experimented with various combinations of embedding types and models, though not all are detailed here due to space constraints (see Table IV). While the sources we referenced used TF-IDF, Bag of Words, and Word2Vec, these did not yield satisfactory results. Instead, sentence embeddings provided the best outcomes across all models, though still below the leaderboard baseline.

Model	Embedding	Validation Accuracy (%)
SVC	Sentence Embedding	77.35
Logistic Regression	Sentence Embedding	77.42
Random Forest	Sentence Embedding	77.58
SVC	BOW	76.97
Random Forest	BOW	75.86

Table IV. Classification accuracy scores on a 10% held-out validation set for the classical ML-based baselines

DIFFERENT BERT EMBEDDINGS

We experimented with various BERT models to use for the input embeddings of *BERTweetConvFusionNet*. The results are summarized in the table V below. *BERTweet* clearly outperformed the other BERT variations.

Model Embedding	Validation Accuracy (%)
DistilBERT Embd	86.86
DistilRoBERTa Embd	81.68
BERTweet Embd	**90.18**

Table V. Classification accuracy scores on *BERTweetConvFusionNet* with different BERT model embeddings